On Fast Convergence of Proximal Algorithms for SQRT-Lasso Optimization: Don't Worry About its Nonsmooth Loss Function

Xinguo Li Princeton University
Haoming Jiang Georgia Institute of Technology
Jarvis Haupt University of Minnesota
Raman Arora Johns Hopkins University
Han Liu Northwestern University
Mingyi Hong University of Minnesota
Tuo Zhao Georgia Institute of Technology

Abstract

Many machine learning techniques sacrifice convenient computational structures to gain estimation robustness and modeling flexibility. However, by exploring the modeling structures, we find these "sacrifices" do not always require more computational efforts. To shed light on such a "free-lunch" phenomenon, we study the square-root-Lasso (SQRT-Lasso) type regression problem. Specifically, we show that the nonsmooth loss functions of SQRT-Lasso type regression ease tuning effort and gain adaptivity to inhomogeneous noise, but is not necessarily more challenging than Lasso in computation. We can directly apply proximal algorithms (e.g. proximal gradient descent, proximal Newton, and proximal quasi-Newton algorithms) without worrying about the nonsmoothness of the loss function. Theoretically, we prove that the proximal algorithms enjoy fast local convergence with high probability. Our numerical experiments also show that when further combined with the pathwise optimization scheme, the proximal algorithms significantly outperform other competing algorithms.

1 INTRODUCTION

Many statistical machine learning methods can be formulated as optimization problems in the following form

$$\min_{\theta} \mathcal{L}(\theta) + \mathcal{R}(\theta), \tag{1.1}$$

where $\mathcal{L}(\theta)$ is a loss function and $\mathcal{R}(\theta)$ is a regularizer. When the loss function is smooth and has a Lipschitz continuous gradient, (1.1) can be efficiently solved by simple proximal gradient descent and proximal Newton algorithms (also requires a Lipschitz continuous Hessian matrix of $\mathcal{L}(\theta)$). Some statistical machine learning methods, however, sacrifice convenient computational structures to gain estimation robustness and modeling flexibility [1, 2, 3]. Taking SVM as an example, the hinge loss function gains estimation robustness, but sacrifices the smoothness (compared with the square hinge loss function). However, by exploring the structure of the problem, we find that these "sacrifices" do not always require more computational efforts.

Advantage of SQRT-Lasso over Lasso. To shed light on such a "free-lunch" phenomenon, we study the high dimensional square-root (SQRT) Lasso regression problem [2, 4]. Specifically, we consider a sparse linear model in high dimensions,

$$y = X\theta^* + \epsilon,$$

where $X \in \mathbb{R}^{n\times d}$ is the design matrix, $y \in \mathbb{R}^n$ is the response vector, $\epsilon \sim N(0, \sigma^2 I_n)$ is the random noise, and θ^* is the sparse unknown regression coefficient vector (all of the following analysis can be extended to the weak sparsity based on [5]). To estimate θ^*, [6] propose the well-known Lasso estimator by solving

$$\overline{\theta}^{\mathsf{Lasso}} = \operatorname*{argmin}_{\theta} \frac{1}{n}\|y - X\theta\|_2^2 + \lambda_{\mathsf{Lasso}}\|\theta\|_1, \tag{1.2}$$

where λ_{Lasso} is the regularization parameter. Existing literature shows that given

$$\lambda_{\mathsf{Lasso}} \asymp \sigma\sqrt{\frac{\log d}{n}}, \tag{1.3}$$

$\overline{\theta}^{\mathsf{Lasso}}$ is minimax optimal for parameter estimation in high dimensions. Note that the optimal regularization parameter for Lasso in (1.3), however, requires the prior knowledge of the unknown parameter σ. This requires the regularization parameter to be carefully tuned over a wide range of potential values to get a good finite-sample performance.

To overcome this drawback, [2] propose the SQRT-Lasso estimator by solving

$$\overline{\theta}^{\mathsf{SQRT}} = \operatorname*{argmin}_{\theta\in\mathbb{R}^d} \frac{1}{\sqrt{n}}\|y - X\theta\|_2 + \lambda_{\mathsf{SQRT}}\|\theta\|_1, \tag{1.4}$$

where λ_{SQRT} is the regularization parameter. They further show that $\overline{\theta}^{\mathsf{SQRT}}$ is also minimax optimal in parameter estimation, but the optimal regularization parameter is

$$\lambda_{\mathsf{SQRT}} \asymp \sqrt{\frac{\log d}{n}}. \tag{1.5}$$

Since (1.5) no longer depends on σ, SQRT-Lasso eases tuning effort.

Extensions of SQRT-Lasso. Besides the tuning advantage, the regularization selection for SQRT-Lasso type methods is also adaptive to inhomogeneous noise. For example, [3] propose a multivariate SQRT-Lasso for sparse multitask learning. Given a matrix $A \in \mathbb{R}^{d\times d}$, let A_{*k} denote the k-th column of A, and A_{i*} denote the i-th row of A. Specifically, [3] consider a multitask regression model

$$Y = X\Theta^* + W,$$

where $X \in \mathbb{R}^{n\times d}$ is the design matrix, $Y \in \mathbb{R}^{n\times m}$ is the response matrix, $W_{*k} \sim N(0, \sigma_k^2 I_n)$ is the random noise, and $\Theta^* \in \mathbb{R}^{d\times m}$ is the unknown row-wise sparse coefficient matrix, i.e., Θ^* has many rows with all zero entries. To estimate Θ^*, [3] propose a calibrated multivariate regression (CMR) estimator by solving

$$\overline{\theta}^{\mathsf{CMR}} = \operatorname*{argmin}_{\theta\in\mathbb{R}^{d\times m}} \frac{1}{\sqrt{n}} \sum_{k=1}^{m} \|Y_{*k} - X\Theta_{*k}\|_2 + \lambda_{\mathsf{CMR}}\|\Theta\|_{1,2},$$

where $\|\Theta\|_{1,2} = \sum_{j=1}^{d} \|\Theta_{j*}\|_2$. [3] further shows that the regularization of CMR approach is adaptive to σ_k's for each regression task, i.e., $Y_{*k} = X\Theta^*_{*k} + W_{*k}$, and therefore CMR achieves better performance in parameter estimation and variable selection than its least square loss based counterpart. With a similar motivation, [7] propose a node-wise SQRT-Lasso approach for sparse precision matrix estimation. Due to space limit, please refer to [7] for more details.

Existing Algorithms for SQRT-Lasso Optimization. Despite of these good properties, in terms of optimization, (1.4) for SQRT-Lasso is computationally more challenging than (1.2) for Lasso. The ℓ_2 loss in (1.4) is not necessarily differentiable, and does not have a Lipschitz continuous gradient, compared with the least square loss in (1.2). A few algorithms have been proposed for solving (1.4) in existing literature, but none of them are satisfactory when n and d are large. [2] reformulate (1.4) as a second order cone program (SOCP) and solve by an interior point method with a computational cost of $\mathcal{O}(nd^{3.5}\log(\epsilon^{-1}))$, where ϵ is a pre-specified optimization accuracy; [8] solve (1.4) by an alternating direction method of multipliers (ADMM) algorithm with a computational cost of $\mathcal{O}(nd^2/\epsilon)$; [4] propose to solve the variational form of (1.4) by an alternating minimization algorithm, and [9] further develop a coordinate descent subroutine to accelerate its computation. However, no iteration complexity is established in [9]. Our numerical study shows that their algorithm only scales to moderate problems. Moreover, [9] require a good initial guess for the lower bound of σ. When the initial guess is inaccurate, the empirical convergence can be slow.

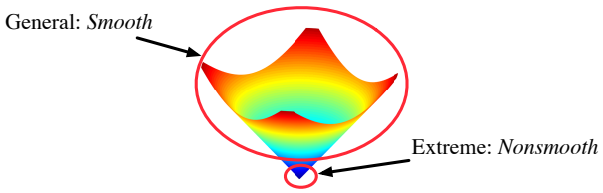

Figure 1: The extreme and general cases of the ℓ_2 loss. The nonsmooth region $\{\theta : y - X\theta = 0\}$ is out of our interest, since it corresponds to those overfitted regression models

Our Motivations. The major drawback of the aforementioned algorithms is that they do not explore the modeling structure of the problem. The ℓ_2 loss function is not differentiable only when the model are overfitted, i.e., the residuals are zero values $y - X\theta = 0$. Such an extreme scenario rarely happens in practice, especially when SQRT-Lasso is equipped with a sufficiently large regularization parameter λ_{SQRT} to yield a sparse solution and prevent overfitting. Thus, we can treat the ℓ_2 loss as an "almost" smooth function. Moreover, our theoretical investigation indicates that the ℓ_2 loss function also enjoys the restricted strong convexity, smoothness, and Hessian smoothness. In other words, the ℓ_2 loss function behaves as a strongly convex and smooth over a sparse domain. An illustration is provided in Figure 1.

Our Contributions. Given these nice geometric properties of the ℓ_2 loss function, we can directly solve (1.4) by proximal gradient descent (Prox-GD), proximal Newton (Prox-Newton), and proximal Quasi-Newton (Prox-Quasi-Newton) algorithms [10, 11]. Existing literature only apply these algorithms to solve optimization problems in statistical machine learning when the loss function is smooth. Our theoretical analysis shows that both algorithms enjoy fast convergence. Specifically, the Prox-GD algorithm achieves a local linear convergence and the Prox-Newton algorithm achieves a local quadratic convergence. The computational performance of these two algorithms can be further boosted in practice, when combined with the pathwise optimization scheme. Specifically, the pathwise optimization scheme solves (1.4) with a decreasing sequence of regularization parameters, $\lambda_0 \geq \ldots \geq \lambda_N$ with $\lambda_N = \lambda_{\mathsf{SQRT}}$. The pathwise optimization scheme helps yield sparse solutions and avoid overfitting throughout all iterations. Therefore, the nonsmooth loss function is differentiable.

Table 1: Comparison with existing algorithms for solving SQRT-Lasso. SOCP: Second-order Cone Programming; TRM: Trust Region Newton; VAM: Variational Alternating Minimization; ADMM: Alternating Direction Method of Multipliers; VCD: Coordinate Descent; Prox-GD: Proximal Gradient Descent; Prox-Newton: Proximal Newton.

	Algorithm	Theoretical Guarantee	Empirical Performance
[2]	SOCP + TRM	$\mathcal{O}(nd^{3.5}\log(\epsilon^{-1}))$	Very Slow
[4]	VAM	N.A.	Very Slow
[8]	ADMM	$\mathcal{O}(nd^2/\epsilon)$	Slow
[9]	VAM + CD	N.A.	Moderate
This paper	Pathwise Prox-GD	$\mathcal{O}(nd\log(\epsilon^{-1}))$	Fast
This paper	Pathwise Prox-Newton + CD	$\mathcal{O}(snd\log\log(\epsilon^{-1}))$	Very Fast

Remark: [9] requires a good initial guess of σ to achieve moderate performance. Otherwise, its empirical performance is similar to ADMM.

Besides sparse linear regression, we extend our algorithms and theory to sparse multitask regression and sparse precision matrix estimation. Extensive numerical results show our algorithms uniformly outperform the competing algorithms.

Key Points of Analysis. We highlight that our local analysis with strong convergence guarantees are novel and nontrivial for solving the SQRT-Lasso problem using simple and efficient proximal algorithms. First of all, sophisticated analysis is required to demonstrate the restricted strong convexity/smoothness and Hessian smoothness of the ℓ_2 loss function over a neighborhood of the underlying model parameter θ^* in high dimensions. These are key properties for establishing the strong convergence rates of proximal algorithms. Moreover, it is involved to guarantee that the output solution of the proximal algorithms do not fall in the nonsmooth region of the ℓ_2 loss function. This is important in guaranteeing the favored computational and statistical properties. In addition, it is technical to show that the pathwise optimization does enter the strong convergence region at certain stage. We defer all detailed analysis to the appendix.

Notations. Given a vector $v \in \mathbb{R}^d$, we define the subvector of v with the j-th entry removed as $v_{\setminus j} \in \mathbb{R}^{d-1}$. Given an index set $\mathcal{I} \subseteq \{1, ..., d\}$, let $\overline{\mathcal{I}}$ be the complementary set to $\mathcal{I}$ and $v_{\mathcal{I}}$ be a subvector of v by extracting all entries of v with indices in $\mathcal{I}$. Given a matrix $A \in \mathbb{R}^{d\times d}$, we denote A_{*j} (A_{k*}) the j-th column (k-th row), $A_{\setminus i\setminus j}$ as a submatrix of A with the i-th row and the j-th column removed and $A_{\setminus ij}$ ($A_{i\setminus j}$) as the j-th column (i-th row) of A with its i-th entry (j-th entry) removed. Let $\Lambda_{\max}(A)$ and $\Lambda_{\min}(A)$ be the largest and smallest eigenvalues of A respectively. Given an index set $\mathcal{I} \subseteq \{1, ..., d\}$, we use $A_{\mathcal{I}\mathcal{I}}$ to denote a submatrix of A by extracting all entries of A with both row and column indices in $\mathcal{I}$. We denote $A \succ 0$ if A is a positive-definite matrix. Given two real sequences $\{A_n\}, \{a_n\}$, we use conventional notations $A_n = \mathcal{O}(a_n)$ (or $A_n = \Omega(a_n)$) denote the limiting behavior, ignoring constant, $\widetilde{\mathcal{O}}$ to denote limiting behavior further ignoring logarithmic factors, and $\mathcal{O}_P(\cdot)$ to denote the limiting behavior in probability. $A_n \asymp a_n$ if $A_n = \mathcal{O}(a_n)$ and $A_n = \Omega(a_n)$ simultaneously. Given a vector $x \in \mathbb{R}^d$ and a real value $\lambda > 0$, we denote the soft thresholding operator $S_\lambda(x) = [\mathrm{sign}(x_j)\max\{|x_j| - \lambda, 0\}]_{j=1}^d$. We use "w.h.p." to denote "with high probability".

2 ALGORITHM

We present the Prox-GD and Prox-Newton algorithms. For convenience, we denote

$$\mathcal{F}_\lambda(\theta) = \mathcal{L}(\theta) + \lambda\|\theta\|_1,$$

where $\mathcal{L}(\theta) = \frac{1}{\sqrt{n}}\|y - X\theta\|_2$. Since SQRT-Lasso is equipped with a sufficiently large regularization parameter λ to prevent overfitting, i.e., $y - X\theta \neq 0$, we treat $\mathcal{L}(\theta)$ as a differentiable function in this section. Formal justifications will be provided in the next section.

2.1 PROXIMAL GRADIENT DESCENT ALGORITHM

Given $\theta^{(t)}$ at t-th iteration, we consider a quadratic approximation of $\mathcal{F}_\lambda(\theta)$ at $\theta = \theta^{(t)}$ as

$$\mathcal{Q}_\lambda(\theta, \theta^{(t)}) = \mathcal{L}(\theta^{(t)}) + \nabla\mathcal{L}(\theta^{(t)})^\top(\theta - \theta^{(t)}) + \frac{L^{(t)}}{2}\|\theta - \theta^{(t)}\|_2^2 + \lambda\|\theta\|_1, \quad (2.1)$$

where $L^{(t)}$ is a step size parameter determined by the backtracking line search. We then take

$$\theta^{(t+1)} = \operatorname*{argmin}_\theta \mathcal{Q}_\lambda(\theta, \theta^{(t)}) = \mathcal{S}_{\frac{\lambda}{L^{(t)}}}\left(\theta^{(t)} - \frac{\nabla\mathcal{L}(\theta^{(t)})}{L^{(t)}}\right).$$

For simplicity, we denote $\theta^{(t+1)} = \mathcal{T}_{L^{(t+1)},\lambda}(\theta^{(t)})$. Given a pre-specified precision ε, we terminate the it-

erations when the approximate KKT condition holds:

$$\omega_\lambda(\theta^{(t)}) = \min_{g \in \partial \|\theta^{(t)}\|_1} \|\nabla \mathcal{L}(\theta^{(t)}) + \lambda g\|_\infty \le \varepsilon. \quad (2.2)$$

2.2 PROXIMAL NEWTON ALGORITHM

Given $\theta^{(t)}$ at t-th iteration, we denote a quadratic term of θ as

$$\|\theta - \theta^{(t)}\|^2_{\nabla^2 \mathcal{L}(\theta^{(t)})} = (\theta - \theta^{(t)})^\top \nabla^2 \mathcal{L}(\theta^{(t)})(\theta - \theta^{(t)}),$$

and consider a quadratic approximation of $\mathcal{F}_\lambda(\theta)$ at $\theta = \theta^{(t)}$ is

$$\mathcal{Q}_\lambda(\theta, \theta^{(t)}) = \mathcal{L}(\theta^{(t)}) + \nabla \mathcal{L}(\theta^{(t)})^\top (\theta - \theta^{(t)}) + \frac{1}{2}\|\theta - \theta^{(t)}\|^2_{\nabla^2 \mathcal{L}(\theta^{(t)})} + \lambda \|\theta\|_1. \quad (2.3)$$

We then take

$$\theta^{(t+0.5)} = \operatorname*{argmin}_\theta \mathcal{Q}_\lambda(\theta, \theta^{(t)}). \quad (2.4)$$

An additional backtracking line search procedure is required to obtain

$$\theta^{(t+1)} = \theta^{(t)} + \eta_t(\theta^{(t+0.5)} - \theta^{(t)}),$$

which guarantees $\mathcal{F}_\lambda(\theta^{(t+1)}) \le \mathcal{F}_\lambda(\theta^{(t)})$. The termination criterion for Prox-Newton is same with (2.2).

Remark 2.1. The ℓ_1 regularized quadratic problem in (2.4) can be solved efficiently by the coordinate descent algorithm combined with the active set strategy. See more details in [12]. The computational cost is $\mathcal{O}(snd)$, where $s \ll d$ is the solution sparsity.

Algorithm 1 Prox-GD algorithm for solving the SQRT-Lasso optimization (1.4). We treat $\mathcal{L}(\theta)$ as a differentiable function.

Input: y, X, λ, ε, $L_{\max} > 0$
Initialize: $\theta^{(0)}$, $t \leftarrow 0$, $L^{(0)} \leftarrow L_{\max}$, $\widetilde{L}^{(0)} \leftarrow L^{(0)}$
Repeat: $t \leftarrow t + 1$
  Repeat: (Line Search)
    $\theta^{(t)} \leftarrow \mathcal{T}_{\widetilde{L}^{(t)}, \lambda}(\theta^{(t-1)})$
    If $\mathcal{F}_\lambda(\theta^{(t)}) < \mathcal{Q}_\lambda(\theta^{(t)}, \theta^{(t-1)})$
    Then $\widetilde{L}^{(t)} \leftarrow \frac{\widetilde{L}^{(t)}}{2}$
  Until: $\mathcal{F}_\lambda(\theta^{(t)}) \ge \mathcal{Q}_\lambda(\theta^{(t)}, \theta^{(t-1)})$
  $L^{(t)} \leftarrow \min\{2\widetilde{L}^{(t)}, L_{\max}\}$, $\widetilde{L}^{(t)} \leftarrow L^{(t)}$
  $\theta^{(t)} \leftarrow \mathcal{T}_{L^{(t)}, \lambda}(\theta^{(t-1)})$
Until: $\omega_\lambda(\theta^{(t)}) \le \varepsilon$
Return: $\widehat{\theta} \leftarrow \theta^{(t)}$

Details of Prox-GD and Prox-Newton algorithms are summarized in Algorithms 1 and 2 respectively. To facilitate global fast convergence, we further combine the pathwise optimization [13] with the proximal algorithms. See more details in Section 4.

Remark 2.2. We can also apply proximal quasi-Newton method. Accordingly, at each iteration, the Hessian matrix in (2.3) is replaced with an approximation. See [14] for more details.

Algorithm 2 Prox-Newton algorithm for solving the SQRT-Lasso optimization (1.4). We treat $\mathcal{L}(\theta)$ as a differentiable function.

Input: y, X, λ, ε
Initialize: $\theta^{(0)}$, $t \leftarrow 0$, $\mu \leftarrow 0.9$, $\alpha \leftarrow \frac{1}{4}$
Repeat: $t \leftarrow t + 1$
  $\theta^{(t)} \leftarrow \operatorname{argmin}_\theta \mathcal{Q}_\lambda(\theta, \theta^{(t-1)})$
  $\Delta\theta^{(t)} \leftarrow \theta^{(t)} - \theta^{(t-1)}$
  $\gamma_t \leftarrow \nabla \mathcal{L}\left(\theta^{(t-1)}\right)^\top \Delta\theta^{(t)} + \lambda\left(\|\theta^{(t)}\|_1 - \|\theta^{(t-1)}\|_1\right)$
  $\eta_t \leftarrow 1$, $q \leftarrow 0$
  Repeat: $q \leftarrow q + 1$ (Line Search)
    $\eta_t \leftarrow \mu^q$
  Until $\mathcal{F}_\lambda\left(\theta^{(t-1)} + \eta_t \Delta\theta^{(t)}\right) \le \mathcal{F}_\lambda\left(\theta^{(t-1)}\right) + \alpha\eta_t\gamma_t$
  $\theta^{(t)} \leftarrow \theta^{(t)} + \eta_t \Delta\theta^{(t-1)}$
Until: $\omega_\lambda(\theta^{(t)}) \le \varepsilon$
Return: $\widehat{\theta} \leftarrow \theta^{(t)}$

3 THEORETICAL ANALYSIS

We start with defining the locally restricted strong convexity/smoothness and Hessian smoothness.

Definition 3.1. Denote

$$\mathcal{B}_r = \{\theta \in \mathbb{R}^d : \|\theta - \theta^*\|_2^2 \le r\}$$

for some constant $r \in \mathbb{R}^+$. For any $v, w \in \mathcal{B}_r$ satisfying $\|v - w\|_0 \le s$, $\mathcal{L}$ is *locally restricted strongly convex* (LRSHC), *smooth* (LRSS), and *Hessian smooth* (LRHS) respectively on $\mathcal{B}_r$ at sparsity level s, if there exist universal constants $\rho_s^-, \rho_s^+, L_s \in (0, \infty)$ such that

$$\begin{aligned}
&\text{LRSC:} \mathcal{L}(v) - \mathcal{L}(w) - \nabla \mathcal{L}(w)^\top (v - w) \ge \frac{\rho_s^-}{2}\|v - w\|_2^2, \\
&\text{LRSS:} \mathcal{L}(v) - \mathcal{L}(w) - \nabla \mathcal{L}(w)^\top (v - w) \le \frac{\rho_s^+}{2}\|v - w\|_2^2, \\
&\text{LRHS:} u^\top (\nabla^2 \mathcal{L}(v) - \nabla^2 \mathcal{L}(w)) u \le L_s \|v - w\|_2^2,
\end{aligned} \quad (3.1)$$

for any u satisfying $\|u\|_0 \le s$ and $\|u\|_2 = 1$. We define the locally restricted condition number at sparsity level s as $\kappa_s = \frac{\rho_s^+}{\rho_s^-}$.

LRSC and LRSS are locally constrained variants of restricted strong convexity and smoothness [15, 16], which are keys to establishing the strong convergence guarantees in high dimensions. The LRHS is parallel to the local Hessian smoothness for analyzing the proximal Newton algorithm in low dimensions [11]. This is also closely related to the self-concordance [17] in the analysis of Newton method [18]. Note that r is associated with the radius of the neighborhood of θ^* excluding the nonsmooth (and overfitted) region of the problem to guarantee strong convergence, which will be quantified below.

Next, we prove that the ℓ_2 loss of SQRT-Lasso enjoys the good geometric properties defined in Definition 3.1 under mild modeling assumptions.

Lemma 3.2. Suppose $\boldsymbol{\epsilon}$ has i.i.d. sub-Gaussian entries with $\mathbb{E}[\epsilon_i] = 0$ and $\mathbb{E}[\epsilon_i^2] = \sigma^2$, $\|\theta^*\|_0 = s^*$. Then for any $\lambda \geq C_1\sqrt{\frac{\log d}{n}}$, w.h.p. we have

$$\lambda \geq \frac{C_1}{4}\|\nabla\mathcal{L}(\theta^*)\|_\infty.$$

Moreover, given each row of the design matrix X independently sampled from a sub-Gaussian distribution with the positive definite covariance matrix $\Sigma_X \in \mathbb{R}^{d\times d}$ with bounded eigenvalues. Then for

$$n \geq C_2 s^* \log d,$$

$\mathcal{L}(\theta)$ satisfies LRSC, LRSS, and LRHS properties on $\mathcal{B}_r$ at sparse level $s^* + 2\widetilde{s}$ with high probability. Specifically, (3.1) holds with

$$\rho^+_{s^*+2\widetilde{s}} \leq \frac{C_3}{\sigma},\ \rho^-_{s^*+2\widetilde{s}} \geq \frac{C_4}{\sigma} \text{ and } L_{s^*+2\widetilde{s}} \leq \frac{C_5}{\sigma},$$

where $C_1, \ldots, C_5 \in \mathbb{R}^+$ are generic constants, and r and $\widetilde{s}$ are sufficiently large constants, i.e., $\widetilde{s} > (196\kappa^2_{s^*+2\widetilde{s}} + 144\kappa_{s^*+2\widetilde{s}})s^*$.

The proof is provided in Appendix A. Lemma 3.2 guarantees that with high probability:

(i) λ is sufficiently large to eliminate the irrelevant variables and yields sufficiently sparse solutions [19, 5];

(ii) LRSC, LRSS, and LRHS hold for the ℓ_2 loss of SQRT-Lasso such that fast convergence of the proximal algorithms can be established in a sufficiently large neighborhood of θ^* associated with r;

(iii) (3.1) holds in $\mathcal{B}_r$ at sparsity level $s^* + 2\widetilde{s}$. Such a property is another key to the fast convergence of the proximal algorithms, because the algorithms can not ensure that the nonzero entries exactly falling in the true support set of θ^*.

3.1 LOCAL LINEAR CONVERGENCE OF PROX-GD

For notational simplicity, we denote

$$\mathcal{S}^* = \{j \mid \theta^*_j \neq 0\},\quad \overline{\mathcal{S}}^* = \{j \mid \theta^*_j = 0\},\ \text{and}$$
$$\mathcal{B}_r^{s^*+\widetilde{s}} = \mathcal{B}_r \cap \{\theta \in \mathbb{R}^d : \|\theta - \theta^*\|_0 \leq s^* + \widetilde{s}\}.$$

To ease the analysis, we provide a local convergence analysis when $\theta \in \mathcal{B}_r^{s^*+\widetilde{s}}$ is sufficiently close to θ^*. The convergence of Prox-GD is presented as follows.

Theorem 3.3. Suppose X and n satisfy conditions in Lemma 3.2. Given λ and $\theta^{(0)}$ such that $\lambda \geq \frac{C_1}{4}\|\nabla\mathcal{L}(\theta^*)\|_\infty$, $\|\theta^{(0)} - \theta^*\|_2^2 \leq s^*\left(8\lambda/\rho^-_{s^*+\widetilde{s}}\right)^2$ and $\theta^{(0)} \in \mathcal{B}_r^{s^*+\widetilde{s}}$, we have sufficiently sparse solutions throughout all iterations, i.e.,

$$\|[\theta^{(t)}]_{\overline{\mathcal{S}}^*}\|_0 \leq \widetilde{s}.$$

Moreover, given $\varepsilon > 0$, we need at most

$$T = \mathcal{O}\left(\kappa_{s^*+2\widetilde{s}}\log\left(\frac{\kappa^3_{s^*+2\widetilde{s}}s^*\lambda^2}{\varepsilon^2}\right)\right)$$

iterations to guarantee that the output solution $\widehat{\theta}$ satisfies

$$\|\widehat{\theta} - \overline{\theta}\|_2^2 = \mathcal{O}\left(\left(1 - \tfrac{1}{8\kappa_{s^*+2\widetilde{s}}}\right)^T \varepsilon\lambda s^*\right) \text{ and}$$
$$\mathcal{F}_\lambda(\widehat{\theta}) - \mathcal{F}_\lambda(\overline{\theta}) = \mathcal{O}\left(\left(1 - \tfrac{1}{8\kappa_{s^*+2\widetilde{s}}}\right)^T \varepsilon\lambda s^*\right),$$

where $\overline{\theta}$ is the unique sparse global optimum to (1.4) with $\|[\overline{\theta}]_{\overline{\mathcal{S}}^*}\|_0 \leq \widetilde{s}$.

The proof is provided in Appendix C. Theorem 3.3 guarantees that when properly initialized, the Prox-GD algorithm iterates within the smooth region, maintains the solution sparsity, and achieves a local linear convergence to the unique sparse global optimum to (1.4).

3.2 LOCAL QUADRATIC CONVERGENCE OF PROX-NEWTON

We then present the convergence analysis of the Prox-Newton algorithm as follows.

Theorem 3.4. Suppose X and n satisfy conditions in Lemma 3.2. Given λ and $\theta^{(0)}$ such that $\lambda \geq \frac{C_1}{4}\|\nabla\mathcal{L}(\theta^*)\|_\infty$, $\|\theta^{(0)} - \theta^*\|_2^2 \leq s^*\left(8\lambda/\rho^-_{s^*+\widetilde{s}}\right)^2$ and $\theta^{(0)} \in \mathcal{B}_r^{s^*+\widetilde{s}}$, we have sufficiently sparse solutions throughout all iterations, i.e.,

$$\|[\theta^{(t)}]_{\overline{\mathcal{S}}^*}\|_0 \leq \widetilde{s}.$$

Moreover, given $\varepsilon > 0$, we need at most

$$T = \mathcal{O}\left(\log\log\left(\frac{3\rho^+_{s^*+2\widetilde{s}}}{\varepsilon}\right)\right)$$

iterations to guarantee that the output solution $\widehat{\theta}$ satisfies

$$\|\widehat{\theta} - \overline{\theta}\|_2^2 = \mathcal{O}\left(\left(\tfrac{L_{s^*+2\widetilde{s}}}{2\rho^-_{s^*+2\widetilde{s}}}\right)^{2^T} \varepsilon\lambda s^*\right) \text{ and}$$
$$\mathcal{F}_\lambda(\widehat{\theta}) - \mathcal{F}_\lambda(\overline{\theta}) = \mathcal{O}\left(\left(\tfrac{L_{s^*+2\widetilde{s}}}{2\rho^-_{s^*+2\widetilde{s}}}\right)^{2^T} \varepsilon\lambda s^*\right),$$

where $\overline{\theta}$ is the unique sparse global optimum to (1.4).

The proof is provided in Appendix D. Theorem 3.4 guarantees that when properly initialized, the Prox-Newton algorithm also iterates within the smooth region, maintains the solution sparsity, and achieves a local quadratic convergence to the unique sparse global optimum to (1.4).

Remark 3.5. Our analysis can be further extended to the proximal quasi-Newton algorithm. The only technical difference is controlling the error of the Hessian approximation under restricted spectral norm.

3.3 STATISTICAL PROPERTIES

Next, we characterize the statistical properties for the output solutions of the proximal algorithms.

Theorem 3.6. Suppose X, and n satisfy conditions in Lemma 3.2. Given $\lambda = C_1\sqrt{\log d/n}$, if the output solution $\widehat{\theta}$ obtained from Algorithm 1 and 2 satisfies the approximate KKT condition,

$$\omega_\lambda(\widehat{\theta}) \leq \varepsilon = \mathcal{O}\left(\frac{\sigma s^* \log d}{n}\right),$$

then we have:

$$\|\widehat{\theta} - \theta^*\|_2 = \mathcal{O}_P\left(\sigma\sqrt{\tfrac{s^*\log d}{n}}\right) \quad \text{and}$$

$$\|\widehat{\theta} - \theta^*\|_1 = \mathcal{O}_P\left(\sigma s^*\sqrt{\tfrac{\log d}{n}}\right).$$

Moreover, we have

$$|\widehat{\sigma} - \sigma| = \mathcal{O}_P\left(\frac{\sigma s^* \log d}{n}\right), \text{ where } \widehat{\sigma} = \frac{\|y - X\widehat{\theta}\|_2}{\sqrt{n}}.$$

The proof is provided in Appendix E. Recall that we use $\mathcal{O}_P(\cdot)$ to denote the limiting behavior in probability. Theorem 3.6 guarantees that the output solution $\widehat{\theta}$ obtained from Algorithm 1 and 2 achieves the minimax optimal rate of convergence in parameter estimation [20, 21]. Note that in the stopping criteria $\omega_\lambda(\widehat{\theta}) \leq \varepsilon$, ε is not a tuning parameter, where $\mathcal{O}\left(\frac{\sigma s^* \log d}{n}\right)$ only serves as an upper bound and we can choose a small ε as desired. This is fundamentally different with the optimal λ_{Lasso} that tightly depends on σ.

4 BOOSTING PERFORMANCE VIA PATHWISE OPTIMIZATION SCHEME

We then apply the pathwise optimization scheme to the proximal algorithms, which extends the local fast convergence established in Section 3 to the global setting[1]. The pathwise optimization is essentially a multistage optimization scheme for boosting the computational performance [13, 16, 12].

Specifically, we solve (1.4) using a geometrically decreasing sequence of regularization parameters

$$\lambda_{[0]} > \lambda_{[1]} > \ldots > \lambda_{[N]},$$

where $\lambda_{[N]}$ is the target regularization parameter of SQRT-Lasso. This yields a sequence of output solutions

$$\widehat{\theta}_{[0]}, \widehat{\theta}_{[1]}, \ldots, \widehat{\theta}_{[N]},$$

also known as the solution path. At the K-th optimization stage, we choose $\widehat{\theta}_{[K-1]}$ (the output solution of the $(K-1)$-th stage) as the initial solution, and solve (1.4) with $\lambda = \lambda_{[K]}$ using the proximal algorithms. This is also referred as the warm start initialization in existing literature [13]. Details of the pathwise optimization is summarized in Algorithm 3. In terms of $\epsilon_{[K]}$, because we only need high precision for the final stage, we set $\epsilon_{[K]} = \lambda_{[K]}/4 \gg \epsilon_{[N]}$ for $K < N$.

Algorithm 3 The pathwise optimization scheme for the proximal algorithms. We solve the optimization problem using a geometrically decreasing sequence of regularization parameters.

Input: y, X, N, $\lambda_{[N]}$, $\varepsilon_{[N]}$

Initialize: $\widehat{\theta}_{[0]} \leftarrow 0$, $\lambda_{[0]} \leftarrow \|\nabla\mathcal{L}(0)\|_\infty$, $\eta_\lambda \leftarrow \left(\frac{\lambda_{[N]}}{\lambda_{[0]}}\right)^{\frac{1}{N}}$

For: $K = 1, \ldots, N$

$\lambda_{[K]} \leftarrow \eta_\lambda \lambda_{[K-1]}$, $\theta^{(0)}_{[K]} \leftarrow \widehat{\theta}_{[K-1]}$, $\varepsilon_{[K]} \leftarrow \varepsilon_{[N]}$

$\widehat{\theta}_{[K]} \leftarrow$ Prox-Alg $\left(y, X, \lambda_{[K]}, \theta^{(0)}_{[K]}, \varepsilon_{[K]}\right)$

End For

Return: $\widehat{\theta}_{[N]}$

As can be seen in Algorithm 3, the pathwise optimization scheme starts with

$$\lambda_{[0]} = \|\nabla\mathcal{L}(0)\|_\infty = \left\|\frac{X^\top y}{\sqrt{n}\|y\|_2}\right\|_\infty,$$

which yields an all zero solution $\widehat{\theta}_{[0]} = 0$ (null fit). We then gradually decrease the regularization parameter, and accordingly, the number of nonzero coordinates gradually increases.

The next theorem proves that there exists an $N_1 < N$ such that the fast convergence of the proximal algorithms holds for all $\lambda_{[K]}$'s, where $K \in [N_1 + 1, .., N]$.

Theorem 4.1. Suppose the design matrix X is sub-Gaussian, and $\lambda_{[N]} = C_1\sqrt{\log d/n}$. For $n \geq C_2 s^* \log d$ and $\eta_\lambda \in (\frac{5}{6}, 1)$, the following results hold:

(I) There exists an $N_1 < N$ such that

$$r > s^*\left(8\lambda_{N_1}/\rho^-_{s^*+\widetilde{s}}\right)^2;$$

(II) For any $K \in [N_1+1, .., N]$, we have $\|\theta^{(0)}_{[K]} - \theta^*\|_2^2 \leq s^*\left(8\lambda_{[K]}/\rho^-_{s^*+\widetilde{s}}\right)^2$, $\theta^{(0)}_{[K]} \in \mathcal{B}_r^{s^*+\widetilde{s}}$ w.h.p.;

(III) Theorems 3.3 and 3.4 hold for all λ_K's, where $K \in [N_1 + 1, .., N]$ w.h.p..

The proof is provided in Appendix G. Theorem 4.1 implies that for all $\lambda_{[K]}$'s, where $K \in [N_1, N_1 + 1, .., N]$, the regularization parameter is large enough for ensuring the solution sparsity and preventing overfitting. Therefore, the fast convergence of proximal algorithms can be guaranteed. For $\lambda_{[0]}$ to $\lambda_{[N_1]}$, we do not have theoretical justification for the fast convergence due to the limit

[1] We only provide partial theoretical guarantees.

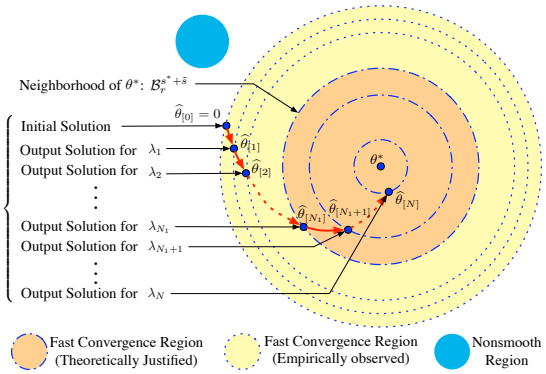

Figure 2: A geometric illustration for the fast convergence of the proximal algorithms. The proximal algorithms combined with the pathwise optimization scheme suppress the overfitting and yield sparse solutions along the solution path. Therefore, the nonsmooth region of the ℓ_2 loss, i.e., the set $\{\theta : y - X\theta = 0\}$, is avoided, and LRSC, LRSS, and LRHS enable the proximal algorithms to achieve fast convergence.

of our proof technique. However, as $\lambda_{[0]}$,..., $\lambda_{[N_1]}$ are all larger than $\lambda_{[N_1+1]}$, we can expect that the obtained model is very unlikely to be overfitted. Accordingly, we can also expect that all intermediate solutions $\widehat{\theta}_{[K]}$'s stay out of the nonsmooth region, and LRSC, LRSS, and LRHS properties should also hold along the solution path. Therefore, the proximal algorithms achieve fast convergence in practice. Note that when the design X is normalized, we have $\lambda_{[0]} = \mathcal{O}(d)$, which implies that the total number N of regularization parameter satisfies

$$N = \mathcal{O}(\log d).$$

A geometric illustration of the pathwise optimization is provided in Figure 2. The supporting numerical experiments are provided in Section 6.

5 EXTENSION TO CMR AND SPME

We extend our algorithm and theory to calibrated multivariate regression (CMR, [3]) and sparse precision matrix estimation (SPME, [7]). Due to space limit, we only provide a brief discussion and omit the detailed theoretical deviation.

Extension to CMR. Recall that CMR solves

$$\overline{\Theta}^{\mathsf{CMR}} = \underset{\theta\in\mathbb{R}^{d\times m}}{\operatorname{argmin}} \frac{1}{\sqrt{n}} \sum_{k=1}^{m} \|Y_{*k} - X\Theta_{*k}\|_2 + \lambda_{\mathsf{CMR}}\|\Theta\|_{1,2}.$$

Similar to SQRT-Lasso, we choose a sufficiently large λ_{CMR} to prevent overfitting. Thus, we can expect

$$\|Y_{*k} - X\Theta_{*k}\|_2 \neq 0 \text{ for all } k = 1, ..., m,$$

and treat the nonsmooth loss of CMR as a differentiable function. Accordingly, we can trim our algorithms and theory for the nonsmooth loss of CMR, and establish fast convergence guarantees, as we discussed in §4.

Extension to SPME. [7] show that a $d \times d$ sparse precision matrix estimation problem is equivalent to a collection of d sparse linear model estimation problems. For each linear model, we apply SQRT-Lasso to estimate the regression coefficient vector and the standard deviation of the random noise. Since SQRT-Lasso is adaptive to imhomogenous noise, we can use one singular regularization parameter to prevent overfitting for all SQRT-Lasso problems. Accordingly, we treat the nonsmooth loss function in every SQRT-Lasso problem as a differentiable function, and further establish fast convergence guarantees for the proximal algorithms combined with the pathwise optimization scheme.

6 NUMERICAL EXPERIMENTS

We compare the computational performance of the proximal algorithms with other competing algorithms using both synthetic and real data. All algorithms are implemented in C++ with double precision using a PC with an Intel 2.4GHz Core i5 CPU and 8GB memory. All algorithms are combined with the pathwise optimization scheme to boost the computational performance. Due to space limit, we omit some less important details.

Synthetic Data: For synthetic data, we generate a training dataset of 200 samples, where each row of the design matrix $X_{i*}$is independently from a 2000-dimensional normal distribution $N(0, \Sigma)$ where $\Sigma_{jj} = 1$ and $\Sigma_{jk} = 0.5$ for all $k \neq j$. We set $s^* = 3$ with $\theta_1^* = 3$, $\theta_2^* = -2$, and $\theta_4^* = 1.5$, and $\theta_j^* = 0$ for all $j \neq 1, 2, 4$. The response vector is generated by $y = X\theta^* + \epsilon$, where ϵ is sampled from $N(0, \sigma^2 I)$.

We first show the **fast convergence** of the proximal algorithms at **every stage** of the pathwise optimization scheme. Here we set $\sigma = 0.5$, $N = 200$, $\lambda_N = \sqrt{\log d/n}$, $\varepsilon_K = 10^{-6}$ for all $K = 1, \ldots, N$. Figure 3 presents the objective gap versus the number of iterations. We can see that the proximal algorithms achieves linear (prox-GD) and quadratic (prox-Newton) convergence at every stage. Since the solution sparsity levels are different at each stage, the slopes of these curves are also different.

Next, we show that the computational performance of the pathwise optimization scheme under different settings. Table 2 presents the timing performance of Prox-GD combined with the pathwise optimization scheme. We can see that $N = 10$ actually leads to better timing performance than $N = 1$. That is because when $N = 1$, the solution path does not fall into the local fast convergence region as illustrated in Figure 2. We can also see that the timing performance of Prox-GD is not sensitive to σ. Moreover, we see that the minimal residual sum of squares along the solution path is much larger than 0, thus the overfitting is prevented and the Prox-GD algo-

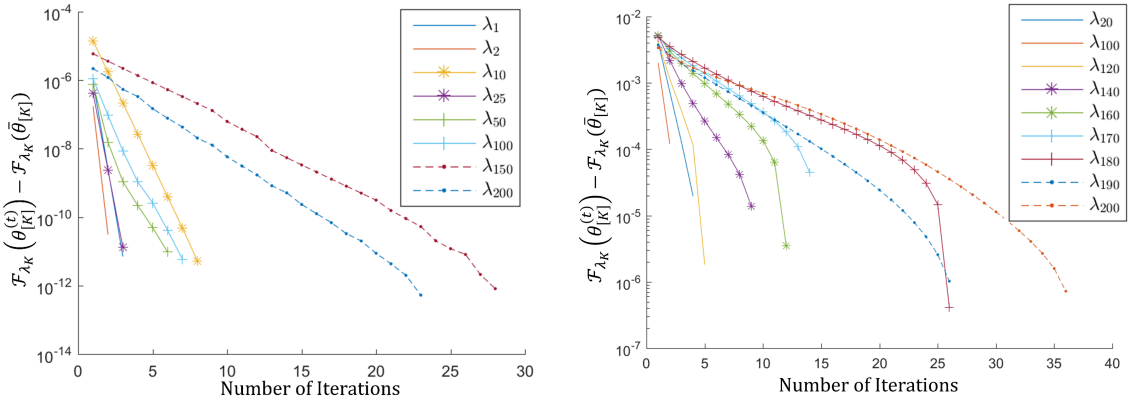

Figure 3: The objective gap v.s. the number of iterations. We can see that the Prox-GD (Left) and Prox-Newton (Right) algorithms achieve linear and quadratic convergence at every stage respectively.

Table 2: Computational performance of Prox-GD on synthetic data under different choices of variance σ, the number of stages N, and the stopping criterion ε_N. The training time is presented, where each entry is the mean execution time in seconds over 100 random trials. The minimal mean square error (MSE) is $\frac{1}{n}\|y - X\widehat{\theta}_{[K]}\|_2^2$, where $\widehat{\theta}_{[K]}$ is the optimal solution that attains $\min \mathcal{F}_{\lambda_K}(\theta)$ for all stages $K = 1, \ldots, N$.

σ	N	ε_N			Minimal MSE	σ	ε_N			Minimal MSE
		10^{-4}	10^{-5}	10^{-6}			10^{-4}	10^{-5}	10^{-6}	
0.1	1	0.372	0.372	0.365	0.013	0.5	0.285	0.295	0.289	0.305
	10	**0.275**	**0.276**	**0.280**			**0.165**	**0.170**	**0.175**	
	30	0.336	0.345	0.351			0.221	0.225	0.228	
1	1	0.235	0.248	0.262	1.183	2	0.432	0.470	0.479	4.220
	10	**0.104**	**0.103**	**0.109**			**0.166**	**0.191**	**0.211**	
	30	0.217	0.222	0.220			0.270	0.296	0.313	

Table 3: Timing comparison between multiple algorithms on real data. Each entry is the execution time in seconds. All experiments are conducted to achieve similar suboptimality.

Data Set	SQRT-Lasso						Lasso
	Prox-GD	Newton	ADMM	ScalReg	CD	Alt.Min	PISTA
Greenhouse	5.812	**1.708**	1027	3181	14.31	99.81	5.113
DrivFace	**0.421**	0.426	18.88	124.0	3.138	17.69	0.414

rithm enjoys the smoothness of the ℓ_2 loss.

Real Data: We adopt two data sets. The first one is the Greenhouse Gas Observing Network Data Set [22], which contains 2921 samples and 5232 variables. The second one is the DrivFace data set, which contains 606 samples and 6400 variables. We compare our proximal algorithms with ADMM in [8], Coordinate Descent (CD) in [9], Prox-GD (solving Lasso) in [16] and Alternating Minimization (Alt.Min.) [4] and ScalReg (a simple variant of Alt. Min) in [23]. Table 3 presents the timing performance of the different algorithms. We can see that Prox-GD for solving SQRT-Lasso significantly outperforms the competitors, and is almost as efficient as Prox-GD for solving Lasso. Prox-Newton is even more efficient than Prox-GD.

Sparse Precision Matrix Estimation. We compare the proximal algorithms with ADMM and CD over real data sets for precision matrix estimation. Particularly, we use four real world biology data sets preprocessed by [24]: Arabidopsis ($d = 834$), Lymph ($d = 587$), Estrogen ($d = 692$), Leukemia ($d = 1,225$). We set three different values for λ_N such that the obtained estimators achieve different levels of sparse recovery. We set $N = 10$, and $\varepsilon_K = 10^{-4}$ for all K's. The timing performance is summarized in Table 4. Prox-GD for solving SQRT-Lasso significantly outperforms the competitors, and is almost as efficient as Prox-GD for solving Lasso. Prox-Newton is even more efficient than Prox-GD.

Calibrated Multivariate Regression. We compare the proximal algorithms with ADMM and CD for CMR on both synthetic data and DrivFace data. For synthetic

Table 4: Timing comparison between multiple algorithms for sparse precision matrix estimation on biology data under different levels of sparsity recovery. Each entry is the execution time in seconds. All experiments are conducted to achieve similar suboptimality. Here CD failed to converge and the program aborted before reaching the desired suboptimality. Scalreg failed to terminate in 1 hour for Estrogen.

Sparsity	Arabidopsis					
	Prox-GD	Newton	ADMM	ScalReg	CD	Alt.Min
1%	5.099	**1.264**	292.0	411.7	12.02	183.6
3%	6.201	**2.088**	339.2	426.1	18.18	217.7
5%	7.122	**2.258**	366.7	435.5	28.60	257.0
Sparsity	Estrogen					
1%	108.2	**3.099**	1597	>3600	136.2	634.1
3%	130.9	**7.101**	1846	>3600	332.0	662.2
5%	143.5	**10.12**	2030	>3600	588.4	739.5
Sparsity	Lymph					
1%	3.709	**0.625**	256.4	354.9	7.208	120.2
3%	4.819	**0.905**	289.1	355.3	10.51	130.6
5%	4.891	**1.123**	310.2	358.7	14.95	148.9
Sparsity	Leukemia					
1%	8.542	**2.715**	331.3	610.1	173.3	239.2
3%	10.56	**3.935**	384.7	766.1	174.3	285.2
5%	10.77	**4.712**	442.5	1274	288.9	333.6

Table 5: Timing comparison between multiple algorithms for calibrated multivariate regression on synthetic and real data with different values of λ_N. Each entry is the execution time in seconds. All experiments are conducted to achieve similar suboptimality. Here CD failed to converge and the program aborted before reaching the desired suboptimality.

λ_N	Synthetic ($\sigma = 1$)				DrivFace			
	Prox-GD	Newton	ADMM	CD	Prox-GD	Newton	ADMM	CD
$\sqrt{\log d/n}$	0.2964	**0.0320**	14.83	2.410	9.562	**0.2186**	158.9	12.77
$2\sqrt{\log d/n}$	0.1725	**0.0213**	2.231	2.227	8.688	**0.1603**	129.4	20.42
$4\sqrt{\log d/n}$	0.0478	**0.0112**	1.868	1.366	1.824	**0.0924**	94.37	19.17

data, the data generating scheme is the same as [3]. Table 5 presents the timing performance. Prox-GD for solving SQRT-Lasso significantly outperforms the competitors, and is almost as efficient as Prox-GD for solving Lasso. Prox-Newton is even more efficient than Prox-GD. CD failed to converge and the program aborted before reaching the desired suboptimality.

7 DISCUSSION AND CONCLUSION

This paper shows that although the loss function in the SQRT-Lasso optimization problem is nonsmooth, we can directly apply the proximal gradient and Newton algorithms with fast convergence. First, the fast convergence rate can be established locally in a neighborhood of θ^*. Note that, due to the limited analytical tools, we are not able to directly extend the analysis to establish a global fast convergence rate. Instead, we resort to the pathwise optimization scheme, which helps establishing empirical global fast convergence for the proximal algorithms as illustrated in Figure 2. Specifically, in the early stage of pathwise scheme, with a large regularization parameter λ, the solution quickly falls into the neighborhood of θ^*, where the problem enjoys good properties. After that, the algorithm can quickly converges to θ^* thanks to the fast local convergence property. Our results corroborate that exploiting modeling structures of machine learning problems is of great importance from both computational and statistical perspectives.

Moreover, we remark that to establish the local fast convergence rate, we prove the restricted strong convexity, smoothness, and Hessian smoothness hold over a neighborhood of θ^*. Rigorously establishing the global fast convergence, however, requires these conditions to hold along the solution path. We conjecture that these conditions do hold because our empirical results show the proximal algorithms indeed achieve fast convergence along the entire solution path of the pathwise optimization. We will look for more powerful analytic tools and defer a sharper characterization to the future effort.

References

[1] L. Wang, "The l1 penalized lad estimator for high dimensional linear regression," *Journal of Multivariate Analysis*, vol. 120, pp. 135–151, 2013.

[2] A. Belloni, V. Chernozhukov, and L. Wang, "Square-root Lasso: pivotal recovery of sparse signals via conic programming," *Biometrika*, vol. 98, no. 4, pp. 791–806, 2011.

[3] H. Liu, L. Wang, and T. Zhao, "Calibrated multivariate regression with application to neural semantic basis discovery." *Journal of Machine Learning Research*, vol. 16, pp. 1579–1606, 2015.

[4] T. Sun and C.-H. Zhang, "Scaled sparse linear regression," *Biometrika*, vol. 99, no. 4, pp. 879–898, 2012.

[5] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu, "A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers," *Statistical Science*, vol. 27, no. 4, pp. 538–557, 2012.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[7] H. Liu, L. Wang *et al.*, "Tiger: A tuning-insensitive approach for optimally estimating gaussian graphical models," *Electronic Journal of Statistics*, vol. 11, no. 1, pp. 241–294, 2017.

[8] X. Li, T. Zhao, X. Yuan, and H. Liu, "The flare package for high dimensional linear regression and precision matrix estimation in R," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 553–557, 2015.

[9] E. Ndiaye, O. Fercoq, A. Gramfort, V. Leclère, and J. Salmon, "Efficient smoothed concomitant lasso estimation for high dimensional regression," in *Journal of Physics: Conference Series*, vol. 904, no. 1. IOP Publishing, 2017, p. 012006.

[10] Y. Nesterov, "Gradient methods for minimizing composite functions," *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, 2013.

[11] J. D. Lee, Y. Sun, and M. A. Saunders, "Proximal newton-type methods for minimizing composite functions," *SIAM Journal on Optimization*, vol. 24, no. 3, pp. 1420–1443, 2014.

[12] T. Zhao, H. Liu, T. Zhang *et al.*, "Pathwise coordinate optimization for sparse learning: Algorithm and theory," *The Annals of Statistics*, vol. 46, no. 1, pp. 180–218, 2018.

[13] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[14] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[15] A. Agarwal, S. Negahban, and M. J. Wainwright, "Fast global convergence rates of gradient methods for high-dimensional statistical recovery," in *Advances in Neural Information Processing Systems*, 2010, pp. 37–45.

[16] L. Xiao and T. Zhang, "A proximal-gradient homotopy method for the sparse least-squares problem," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1062–1091, 2013.

[17] A. Nemirovski, "Interior point polynomial time methods in convex programming," *Lecture Notes*, 2004.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.

[19] P. J. Bickel, Y. Ritov, and A. B. Tsybakov, "Simultaneous analysis of Lasso and Dantzig selector," *The Annals of Statistics*, vol. 37, no. 4, pp. 1705–1732, 2009.

[20] G. Raskutti, M. J. Wainwright, and B. Yu, "Minimax rates of estimation for high-dimensional linear regression over-balls," *Information Theory, IEEE Transactions on*, vol. 57, no. 10, pp. 6976–6994, 2011.

[21] F. Ye and C.-H. Zhang, "Rate minimaxity of the Lasso and Dantzig selector for the ℓ_q loss in ℓ_r balls," *The Journal of Machine Learning Research*, vol. 11, pp. 3519–3540, 2010.

[22] D. D. Lucas, C. Yver Kwok, P. Cameron-Smith, H. Graven, D. Bergmann, T. P. Guilderson, R. Weiss, and R. Keeling, "Designing optimal greenhouse gas observing networks that consider performance and cost," *Geoscientific Instrumentation, Methods and Data Systems*, vol. 4, no. 1, pp. 121–137, 2015. [Online]. Available: https://www.geosci-instrum-method-data-syst.net/4/121/2015/

[23] T. Sun and M. T. Sun, "Package 'scalreg'," 2013.

[24] L. Li and K.-C. Toh, "An inexact interior point method for ℓ_1-regularized sparse covariance selection," *Mathematical Programming Computation*, vol. 2, no. 3-4, pp. 291–315, 2010.

[25] M. Wainwright, "High-dimensional statistics: A non-asymptotic viewpoint," *preparation. University of California, Berkeley*, 2015.

[26] M. Rudelson and S. Zhou, "Reconstruction from anisotropic random measurements," *Information Theory, IEEE Transactions on*, vol. 59, no. 6, pp. 3434–3447, 2013.

[27] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[28] G. Raskutti, M. J. Wainwright, and B. Yu, "Restricted eigenvalue properties for correlated Gaussian designs," *The Journal of Machine Learning Research*, vol. 11, no. 8, pp. 2241–2259, 2010.

[29] Y. Ning, T. Zhao, and H. Liu, "A likelihood ratio framework for high-dimensional semiparametric regression," *The Annals of Statistics*, vol. 45, no. 6, pp. 2299–2327, 2017.

[30] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer, 2004, vol. 87.

[31] P. Bühlmann and S. Van De Geer, *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[32] J. Fan, H. Liu, Q. Sun, and T. Zhang, "I-lamm for sparse learning: Simultaneous control of algorithmic complexity and statistical error," *Annals of statistics*, vol. 46, no. 2, p. 814, 2018.